GISC 4317/6317: GeoComputation/Computer Programming For GIS

**Instructor: Dr. Bryan Chastain**

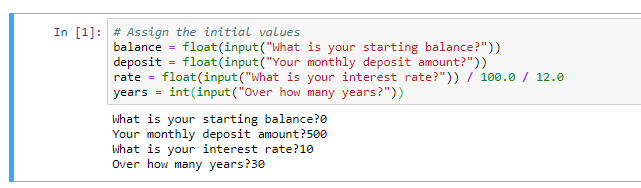
# Lab 2: Control Structure

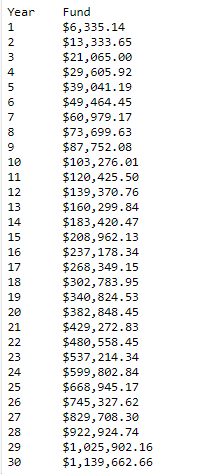
### Part 1. Create a Retirement Fund Calculator

In this exercise, you will create a simple application that calculates retirement funds. The application will allow you to enter a beginning balance, monthly deposit, and annual interest rate, and it will then calculate the total funds available at retirement for the number of years you specify. You will modify the calculator to calculate how many years it will take you to become a millionaire. This exercise assumes that deposits are compounded monthly.

## Step 1. View the finished application

* Open the Retirement Fund Calculator application, which can be found as attached exe file as **lab2soln-python3.exe**





* Enter a beginning balance, monthly deposit, annual interest rate, and number of years until you retire. After you finish entering information, press Enter.
* Your total retirement fund is displayed at the bottom. The first table displays the retirement funds for each year at the specified interest rate. The second table to displays the total retirement funds for the specified number of years for a range of interest rates (Do not worry about this table at this point; you can implement this table for extra credit). In this exercise, you will write the code for this application.
* Type “Y” when asked "Calculate length until you to become a millionaire? (Y/N)". You will see the number of years it will take you to become a millionaire. For your homework, your will need to implement this button.
* Close the application.

**Step 2. Open Jupyter Notebook**

* Open Jupyter and create a new Python 3 Notebook. Save the notebook in your directory as retirement.ipynb.

**Step 3. Calculate Retirement Fund**

Now you'll write the code to calculate the retirement fund.

* You need to assign user inputs to their corresponding variables. Because the input() method returns a string, you have to use a function to convert the string to a number--either float, or integer in this case. Enter these in the first code cell (you may want to have some markdown above it):

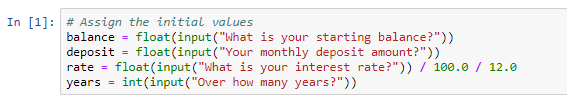
*# Assign the initial values*

balance = float(input("What is your starting balance?"))

deposit = float(input("Your monthly deposit amount?"))

rate = float(input("What is your interest rate?")) / 100.0 / 12.0

years = int(input("Over how many years?"))



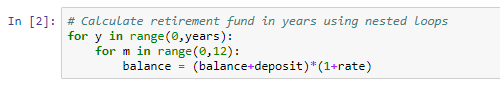
* float(), and int() are data type conversion functions. float() converts the data to a floating point number, and int() converts the data to Integer.
* Most of the above code is straightforward. In the third statement, the annual interest rate is divided by 12 to give the monthly interest rate. *Why do you think the “.0” is included on 100 and 12?*
* Now you are ready to do some calculating. You will use a nested For loop to do the calculation in your next code cell:

*# Calculate retirement fund in years using nested loops*

**for** y **in** range(0,years):

**for** m **in** range(0,12):

balance = (balance+deposit)\*(1+rate)



* In the above code, the inner loop calculates the monthly retirement fund as the fund plus the monthly deposit, and the interest for both of them. The outer loop calculates the retirement funds from the current year to the year when you retire.
* The reason we start the loops at zero is that For loops Python stop **before** they reach the last value. In other words, our inner loop will go from zero to 11, not 12. We start at zero to ensure we get the full range of values we need.
* Note: The statement within the inner loop is a one-line statement.
* Type the above code into the Code Editor. Finally, we output the final balance in a currency format, with commas and two decimal places:

*# Display the result*

**print** ("Retirement Funds: ${:,.2f}".format(balance))



* Type the above code into your cell below the loops. Be sure not to indent before typing. The “{:,.2f}” tells Python that we want it to be comma-separated (,) and have two decimal places (.2f).  
  Re-run the cell.

(*For more on the Python string formatting mini-language, read the documentation:*

[*https://docs.python.org/3.6/library/string.html#format-string-syntax*](https://docs.python.org/3.6/library/string.html#format-string-syntax)*)*

**Step 4. Display the retirement fund for a range of years**

* If you want to see what your retirement fund will be in 10, 20, or 30 years, you can change the Number of Years input to the corresponding number; however, this is not very convenient. Instead, you can adjust your program to display the fund values for a range of years in a tabular format.
* We will create a table with two columns, Year and Fund, separated with a tab. Add the following code just before the For loops:

**print** ("Year\tFund")

* This code concatenates the string Year, a tab, and the string Fund.
* You want to display the fund value for each year, so the fund value should be collected in the outer loop. Inside the nested For loop you added in Step 3, after the line "balance = (balance+deposit)\*(1+rate)", add the following code, set to the same tab spacing as the “for m…” line:

**print** (str(y+1) + "\t${:,.2f}".format(balance))



* This line of code writes to the next line of the “Fund by Years” table including the year and the fund of the year, separated by a tab. The line executes once for each year. Note that we have to add one (y+1) to the output because we started the y loop at zero.
* Before you run the application, you want to modify the label to indicate the interest rate. Add the following code before the “Year\tFund” column headers.



* Note that we have to convert the rate variable to a string in order for it to be concatenated with the rest of the line.
* Run the application to see the retirement fund displayed for a range of years.

**Homework:**

**Undergraduate (GISC 4317)**

* Implement the “When will I become a Millionaire?” option to calculate how many years it will take you to become a millionaire. You should prompt the user as to whether they want to calculate it or not, as in the demo program.

Submit your notebook file (.ipynb) with your name in comments at the top to eLearning.

**Graduate (GISC 6317)**

* Implement the “When will I become a Millionaire?” option to calculate how many years it will take you to become a millionaire. You should prompt the user as to whether they want to calculate it or not, as in the demo program.
* Also, implement the “Fund by interest rate” table as shown in the demo program.
* Check your program against the PEP8 online checker to ensure you’re writing consistent & readable Python code: http://pep8online.com/

Submit your notebook file (.ipynb) with your name in comments at the top to eLearning.

Tip, for both: Write out the pseudo-code for what the script should do, step-by-step before starting on the actual Python code.